Malware

- 20.sept.2008

Chatham House Rule

- When a meeting, or part thereof, is held under the Chatham House Rule, participants are free to use the information received, but neither the identity nor the affiliation of the speaker(s), nor that of any other participant, may be revealed

Malware

- Malicious code often masquerades as good software or attaches itself to good software
- Some malicious programs need host programs
  - Trojan horses, logic bombs, viruses
- Others can exist and propagate independently
  - Worms, automated viruses
- There are many infection vectors and propagation mechanisms
- Really all depends on intentions !!

Trojan Horses

- A trojan horse is malicious code hidden in an apparently useful host program
- When the host program is executed, trojan does something harmful or unwanted
  - Must be user tricked into executing the host program ?
- Trojans do not replicate
  - Main difference from worms and viruses, but today many trojans are spread by virus-like mechanisms

Viruses

- Virus propagates by infecting other programs
  - Automatically creates copies of itself, but to propagate, a human has to run an infected program
    - Self-propagating malicious programs are usually called worms
- Many propagation methods
  - Insert a copy into every executable (.COM, .EXE)
  - Insert a copy into boot sectors of disks
    - “Stoned” virus infected PCs booted from infected floppies, stayed in memory and infected every floppy inserted into PC
  - Infect TSR (terminate-and-stay-resident) routines
    - By infecting a common OS routine, a virus can always stay in memory and infect all disks, executables, etc.

Virus Techniques

- Macro viruses
  - A macro is an executable program embedded in a word processing document (MS Word) or spreadsheet (Excel)
  - When infected document is opened, virus copies itself into global macro file and makes itself auto-executing (e.g., gets invoked whenever any document is opened)
- Stealth techniques
  - Infect OS so that infected files appear normal
    - Used by rootkits (we'll look at them later)
  - Mutate, encrypt parts of code with random key

Stealth Techniques

- Mutation: virus has multiple binary variants
  - Defeats naïve signature-based detection
  - Used by the most successful (i.e., widespread) viruses
    - Tanked: 62 variants, SdDrop: 14 variants
- Aliasing: virus places its copies under different names into the infected host's sharing folder
  - "ICQ Lite .exe", "ICQ Pro 2003b.exe", "MSN Messenger 5.2.exe"

Propagation via Websites

- Websites with popular content
  - Games: 60% of websites contain executable content, one-third contain at least one malicious executable
  - Celebrities, adult content, everything except news
- Large variety of malware
  - But most of the observed programs are variants of the same few adware applications (e.g., WhenU)

Malicious Functionality

- Adware
  - Display unwanted pop-up ads
- Browser hijackers
  - Modify home page, search tools, redirect URLs
- Trojan downloaders
  - Download and install additional malware
- Dialer (expensive toll numbers)
- Keylogging

Drive-By Downloads

- Website “pushes” malicious executable to user’s browser with inline Javascript or pop-up window
  - Naive user may click “Yes” in the dialog box
- Can also install malicious software automatically by exploiting bugs in the user’s browser
  - 1.5% of URLs crawled in the Moshchuk et al. study
- Constant change
  - Many infectious sites exist only for a short time or change substantially from month to month
  - Many sites behave non-deterministically

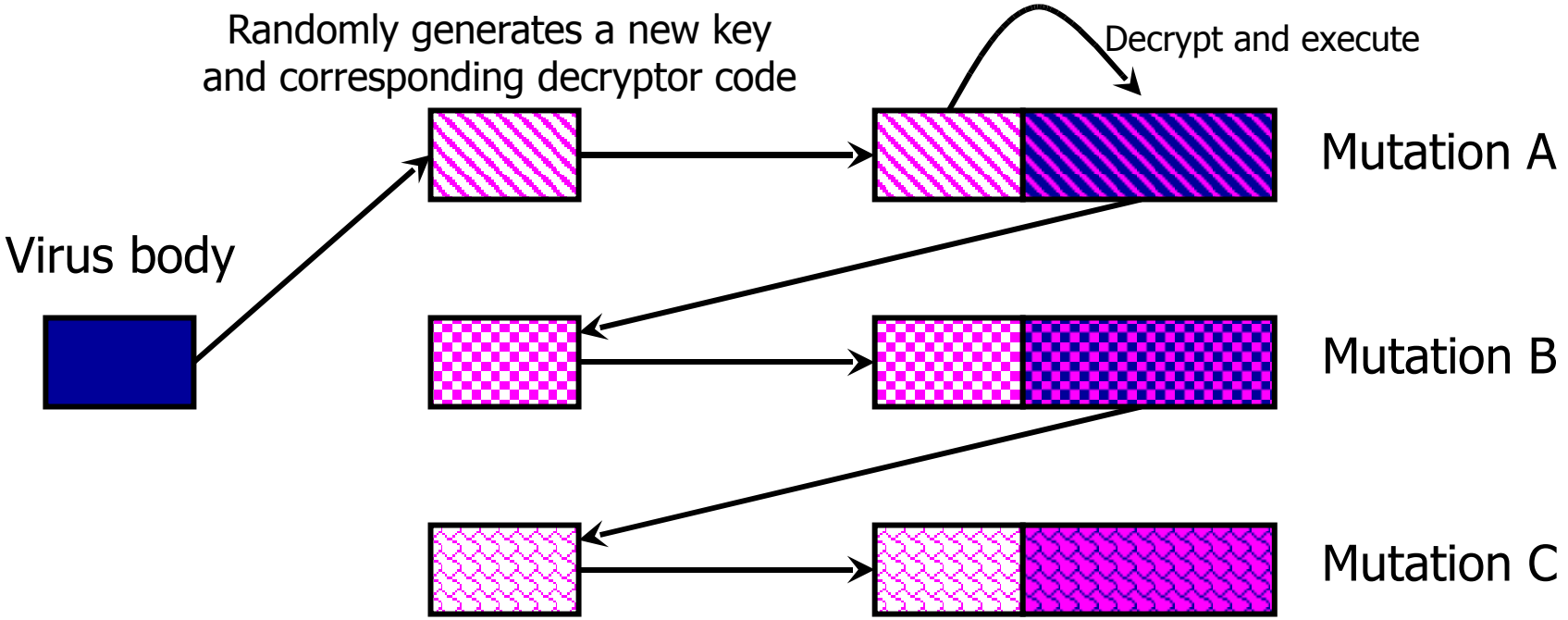
Polymorphic Viruses

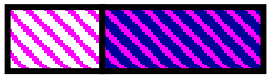
- Encrypted viruses: virus consists of a constant decryptor, followed by the encrypted virus body
  - Relatively easy to detect because decryptor is constant
- Polymorphic viruses: constantly create new random encryptions of the same virus body
  - Marburg (Win95), HPS (Win95), Coke (Win32)
  - Virus includes an engine for creating new keys and new encryptions of the virus body
    - Crypto (Win32) decrypts its body by brute-force key search to avoid explicit decryptor code
  - Decryptor can start with millions of NOPs to defeat emulation

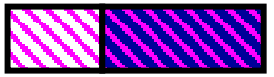
Anti-Virus Technologies

- Simple anti-virus scanners
  - Look for signatures (fragments of known virus code)
  - Heuristics for recognizing code associated with viruses
    - Polymorphic viruses often use decryption loops
  - Integrity checking to find modified files
    - Record file sizes, checksums, MACs (keyed hashes of contents)
    - Often used for rootkit detection (we'll see TripWire later)
- Generic decryption and emulation
  - Emulate CPU execution for a few hundred instructions, virus will eventually decrypt, can recognize known body
    - Does not work very well against mutating viruses and viruses not located near beginning of infected executable

Virus Detection by Emulation



To detect an unknown mutation  of a known virus  ,

emulate CPU execution of  until the current sequence of instruction opcodes matches the known sequence for virus body 

Metamorphic Viruses

- Obvious next step: mutate the virus body, too!
- Virus can carry its source code (which deliberately contains some useless junk) and recompile itself
  - Apparition virus (Win32)
  - Virus first looks for an installed compiler
    - Unix machines have C compilers installed by default
  - Virus changes junk in its source and recompiles itself
    - New binary mutation looks completely different!
- Mutation is common in macro and script viruses
  - Macros/scripts are usually interpreted, not compiled

Obfuscation and Anti-Debugging

- Common in worms, viruses, bots
- Goal: prevent analysis of code and signature-based detection; foil reverse-engineering
  - Insert garbage opcodes and change control structure
  - Different code in each instance
    - Effect of code execution is the same, but difficult to detect by passive analysis
  - Packed binaries
- Detect debuggers and virtual machines, terminate execution

Mutation / Obfuscation Techniques

- Same code, different register names
  - Regswap (Win32)
- Same code, different subroutine order
  - BadBoy (DOS), Ghost (Win32)
  - If n subroutines, then n! possible mutations
- Decrypt virus body instruction by instruction, push instructions on stack, insert and remove jumps, rebuild body on stack
  - Zmorph (Win95)
  - Can be detected by emulation because the rebuilt body has a constant instruction sequence

Mutation Engines

- Real Permutating Engine/RPME, ADMutate, etc.
- Large set of obfuscating techniques
  - Instructions are reordered, branch conditions reversed
  - Jumps and NOPs inserted in random places
  - Garbage opcodes inserted in unreachable code areas
  - Instruction sequences replaced with other instructions that have the same effect, but different opcodes
    - Mutate SUB EAX, EAX into XOR EAX, EAX or

      PUSH EBP; MOV EBP, ESP into PUSH EBP; PUSH ESP; POP EBP
- There is no constant, recognizable virus body!

Example of Zperm Mutation

- From Szor and Ferrie, “Hunting for Metamorphic”

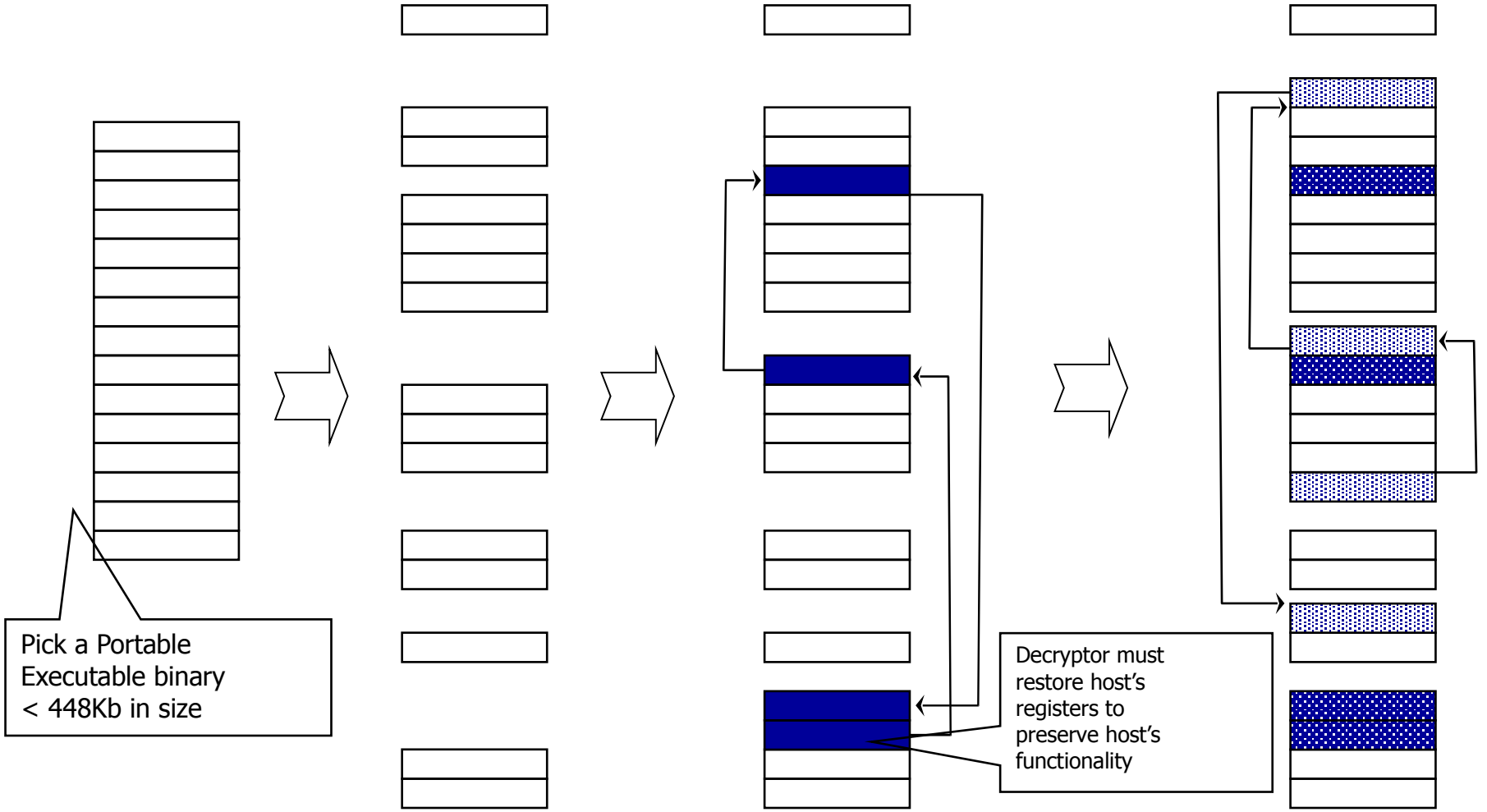
Putting It All Together: Zmist

- Zmist was designed in 2001 by Russian virus writer Z0mbie of “Total Zombification” fame
- New technique: code integration
  - Virus merges itself into the instruction flow of its host
  - “Islands” of code are integrated into random locations in the host program and linked by jumps
  - When/if virus code is run, it infects every available portable executable
    - Randomly inserted virus entry point may not be reached in a particular execution

MISTFALL Disassembly Engine

- To integrate itself into host's instruction flow, virus must disassemble and rebuild host binary
  - See overview at *http://vx.netlux.org/lib/vzo21.html*
- This is very tricky
  - Addresses are based on offsets, which must be recomputed when new instructions are inserted
  - Iterative process: rebuild with new addresses, see if branch destinations changed, then rebuild again
    - Requires 32MB of RAM and explicit section names (DATA, CODE, etc.) in the host binary – doesn't work with every file

Simplified Zmist Infection Process



Disassemble, insert space for new code blocks, generate new binary

Insert mutated virus body

- Split into jump-linked "islands"
- Mutate opcodes (XOR↔SUB, OR↔TEST)
- Swap register moves and PUSH/POP, etc.

Encrypt virus body by XOR (ADD, SUB) with a randomly generated key, insert mutated decryptor

How Hard Is It to Write a Virus?

- 2268 matches for “virus creation tool” in CA’s Spyware Information Cente
  - Including dozens of poly- and metamorphic engines
- OverWritting Virus Construction Toolkit
  - "The perfect choice for beginners”
- Biological Warfare Virus Creation Kit
- Vbs Worm Generator (for Visual Basic worms)
  - Used to create the Anna Kournikova worm
- Many others
- http://www.youtube.com/watch?v=RGTM-WZnQ
-

Home reading

www.symantec.com/avcenter/reference/hunting.for.metamorphic.pdf